

Northumbria Research Link

Citation: Clement, Paul and Vickers, Paul (2002) From Server to PDA: An HCI Perspective on Porting Wireless Roaming Business Applications. In: IWNA5: 5th IEEE International Workshop on Networked Appliances, 30-31 October 2002, Liverpool.

URL:

This version was downloaded from Northumbria Research Link:
<http://nrl.northumbria.ac.uk/id/eprint/11266/>

Northumbria University has developed Northumbria Research Link (NRL) to enable users to access the University's research output. Copyright © and moral rights for items on NRL are retained by the individual author(s) and/or other copyright owners. Single copies of full items can be reproduced, displayed or performed, and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided the authors, title and full bibliographic details are given, as well as a hyperlink and/or URL to the original metadata page. The content must not be changed in any way. Full items must not be sold commercially in any format or medium without formal permission of the copyright holder. The full policy is available online: <http://nrl.northumbria.ac.uk/policies.html>

This document may differ from the final, published version of the research and has been made available online in accordance with publisher policies. To read and/or cite from the published version of the research, please visit the publisher's website (a subscription may be required.)



**Northumbria
University**
NEWCASTLE



UniversityLibrary

From Server to PDA: an HCI Perspective on Porting Wireless Roaming Business Applications

P. CLEMENT
P. VICKERS*

Northumbria University, School of Informatics
Ellison Place, Newcastle-upon-Tyne
NE1 8ST, UK

*paul.vickers@northumbria.ac.uk

Abstract- Advances in processor design, display technology, and lithium polymer batteries have led to a new generation of powerful, compact, and highly mobile devices. There is a range of different physical interfaces available, from the original Palm Pilots and Psion Organisers to the new generation of pocket PCs and personal digital assistants (PDA's). This paper focuses on the stylus/touch-screen-based Compaq iPaq PDA.

Combining a PDA with wireless communication technology has the potential for extending the application of computing in business. Many businesses already use handheld data recording devices. A drawback is the need to return the device to a computer to upload the information into the system where it can be processed and made available to other applications. Using wireless devices this delay can be removed, providing a more accurate and up-to-date data repository.

In order to see how a desktop to PDA migration might be carried out, a prototype interface to a mainstream accounting and stock control system was built to run on a Compaq iPaq 3850. Some of the problem areas of migrating server/desktop functionality to handheld devices, including wireless communication, device limitations, and security were demonstrated and explored with this interface. The informal feedback from the users will be fed into a redeveloped version of the system which will then be evaluated using a formal experiment. The main lesson learnt was that providing desktop/server functionality on a mobile device requires more than copying a system's existing functionality and shrinking the interface to fit on a smaller screen. User feedback from a prototype system supported the idea that applications benefit from a redesign process when being ported to a mobile environment.

I. INTRODUCTION

Devices like the iPaq are compact, highly mobile, feature-rich, and relatively inexpensive. When coupled with wireless network components, GPRS telephone components, and a range of attachments from foldaway keyboards to light-pens, they can be used for many applications. The hardware/software platforms to provide businesses with truly mobile applications are now available. PDAs typically use touch screen technology, and are normally based on a 240x320 pixel display. Commands are executed by touching the display with a finger or stylus thus providing a very direct interaction. Users are often untrained and interaction will need to be simple and fast.

We wished to explore whether a PDA could provide a usable and useful interface to a real business application. Before undertaking formal experimentation, a prototype system was built which was informally tested by users. The lessons learnt from this work will feed into the next stage of the project in which a redesigned system will be built and evaluated in a formal experimental manner.

A scenario to illustrate this is a warehouse worker needing to extract stock from the warehouse to fulfil an order. They would normally print a picking list and walk round the warehouse picking stock from the shelves. Upon returning to a central point they would then update the stock control system with details of the picked stock (or even pass the annotated picking list onto a clerk who would update the system). With a PDA they could update the system as items are picked, and perhaps update the main database to indicate stock items that could not be found so that a purchase order is raised with the supplier.

II. THE PROTOTYPE

A prototype system was built to provide a platform for the evaluation of an interface based on a PDA. The prototype's main purpose was to provide enough functionality to allow an heuristic evaluation of the potential of a handheld device to function as an extension of a mainstream accounting product in order to provide true mobility to a user of the accounting system. The prototype should be able to demonstrate areas of strengths and weaknesses in user interaction of an interface used on such a device. The popular Sage Line 100 system was installed on a laptop computer equipped with a Dell TrueMobile wireless network card to act as the desktop/server system. The prototype was installed on a Compaq iPaq 3850 with a Compaq WL110 wireless PC card. For third party products Line 100's database is accessed via a proprietary ODBC driver with SQL, and this was the method used in this work.

There are a several standards for wireless communication that could be considered for this project.

A. Bluetooth

Bluetooth is an RF specification for short-range point to point and point to multipoint voice and data transfer supporting 128 bit encryption. It provides a 10 metre radius sphere of influence for RF devices and allows for up to eight devices to be connected in a 'piconet'. Ten piconets can be present in one sphere. Bluetooth uses the 2.4GHz Industrial, Scientific and Medical (ISM) frequency band, which is available worldwide, and does not require line of sight for communication. It is generally used for the wireless connection of peripherals (keyboard, mouse, printer etc) though it is also used to provide networking where it is difficult or impossible to use a wired network, for example in a shop or a listed building. Data transfer rates are comparatively slow at 1-2 megabits per second.

B. 802.11b

802.11b allows wireless roaming communication between up to 128 devices in one network segment over a local area. It supports

direct sequence, frequency hopping and diffuse infrared technologies and the standard defines protocols for both ad-hoc and client/server networks. Like Bluetooth, 802.11b devices use the 2.4GHz range, as this requires no transmission licences. Unlike Bluetooth however, 802.11b is rated at 2-11 megabits per second and has a range of 30 metres. This is currently the most popular standard for wireless networks.

The 2.4GHz frequency is also the resonant frequency of water molecules, and therefore the operating frequency of microwave ovens. This makes microwave ovens a potential interference source for devices using this frequency, although ovens are supposed to be shielded to prevent this. Another area of concern is the glare-reducing coating of office windows. Some of these coatings are RF opaque, eliminating signals from one office block to another. There are solutions to this problem, such as placing antennas on roofs.

C. Infrared Data Association (IrDA)

IrDA can achieve transfer rates of up to 4 megabits per second, but is limited to line of sight operations. Laptop computers, handheld devices and some mobile phones have IrDA capability built-in. IrDA does not cause interference with other wireless communications, and its short range of 3 metres makes eavesdropping difficult.

Because of the application environment (a warehouse) we decided to use 802.11b technology which provides a longer range.

A. Development constraints

The Microsoft Pocket PC2002 operating system installed on the iPaq includes the Pocket Internet Explorer browser. The intention was to use Active Server Pages (ASP) technology to produce the interface. However, a problem with the web server software left the server unable to render the ASP code. Instead, the software was coded with TCP/IP sockets and a direct connection between the laptop and the PDA was made (as opposed to an HTTP connection)¹. A server component to handle communication and a client component for the PDA to issue communication requests and to interpret responses from the server component were built. This solution is not ideal as it involves deploying software components to the client, whereas a web-based solution would have removed this requirement, and thus reduced the system's complexity.

The development environments available were limited by the ability of the iPaq and its software. The client and server applications were both written in the appropriate versions of Visual Basic (i.e. VB6 for the server and Embedded VB for the client). However, this led to some compatibility issues. Embedded VB does not support the same range of objects, methods, or properties as VB6. Furthermore, the syntax for common methods and properties is different.

These differences caused frustration and cost time. Experienced developers rely on their memory to allow them to achieve high throughput. They know the structure for all of the more common functions, and only fall back on help systems when the immediate programming problem calls for a solution outside the norm. When using a familiar programming language, changes in the language's

syntax slow the development down; this in turn exacerbates frustration levels.

One other unforeseen problem with developing embedded systems is the available error handling. Embedded VB has only limited error handling facilities. Unlike VB6 a program written in Embedded VB cannot catch and process errors; it can either carry on the processing with invalid data, or stop altogether.

III. DESIGN CONSIDERATIONS

A. Application mapping

Although handheld devices can access and manipulate data like a conventional computer, existing server/desktop applications are not well suited to run on these devices. For using mobile communication devices to access virtual meeting places, Rist [6] used a middle tier component to provide mapping functionality so that the interface components of an application could be used on any device, with the aim of reducing the complexity of the interface to suit the mobile device, such as a PDA or mobile phone. Vanderdonckt et al [7] proposed model-based design allowing different devices to be supported by one application. Again, these techniques involved mapping the functionality and interface components of an application across a range of mobile devices. This is essentially the idea behind the Wireless Application Protocol (WAP) which allows a web based application designed and written for an 800x600 resolution screen to be stripped and rendered on a 240x320 display.

However, much of the information provided by the server application (Fig. 1) is not needed by a person picking an order. Fig. 4 shows how the application was redesigned specifically for use on a small touch screen instead of simply shrinking a larger area.

B. Context Mapping

A context-sensitive application can help users to feel in control of the task. For example, a system that reacts to the current location of the user could show those items closest to them, or even directing them to the next item. Another benefit of context sensitive applications is that other information not necessarily related to the current task could be required. For example, if a store person using a context sensitive device were a great distance from the office, and someone in the office needed urgent information about a product that was close to the device's location, the store person could check on the stock and report back using the device. This may interrupt the task in hand, but could provide more urgent information much faster than previously possible. Likewise the device can be used to provide information to the user about the current task, such as a new delivery of stock that has arrived that can allow the current picking list to be completed. Iacucci [4] asserts that new mobile communication services are likely to be increasingly bonded to the context of the user activities.

¹ The problem has since been resolved, but too late for the current work. It was caused by an unregistered dynamic link library (DLL) component when installing the development environment on laptop that provided the server environment.

Item	Depth	Search This List
1. 1001 F. 1001/1001/1001		
2. 1001 F. 1001/1001/1001		
3. 1001 F. 1001/1001/1001		
4. 1001 F. 1001/1001/1001		

In Parcel Blue Not Count	
In Stock	71 0000
Free Stock	40 0000
Ordered	
Delivered	
To Follow	

Fig. 1. Line 100 picking list

IV. INTERACTION METHODS AND COMPONENTS

A. Drag and drop

The small size of the PDA screen makes the stylus-based drag and drop interface (like the one in the Windows Solitaire game) well suited to the manipulation of objects. On desktop displays the drop zone is often off the display area and the mouse tends to run off the mat before the drop zone is reached, forcing the user to reposition it. By contrast, the entire PDA screen area can be accessed with a single stylus movement. However, this type of interface is harder to implement [3]. Embedded VB does not even support it and Windows API calls must be made instead.

In the drag and drop model the objects being manipulated could be stock codes. For the application of processing a picking list it is unusual for the user to have to enter new stock codes. Therefore, the stock code itself could be dragged and dropped onto an action zone which could be the word 'Picked', or 'OUT' (out of stock). This would help to create what Norman and Draper described as "...the sensation in the user of acting upon the objects of the task domain themselves" [5].

Of course, such an interface may not be so suitable when a finger is used instead of a stylus; more of the screen will be obstructed during the drag operation with the potential for hiding information required to complete the drop operation.

B. Forms, captions, and titles

A PDA screen has around 16% of the area of an 800x600 display. There is simply not enough space on a PDA screen for form captions and frame titles. The interface should be designed so as to eliminate the need for these text areas. Fig. 2 shows that the a form's title area takes up considerable screen real estate. A balance must be struck between information density and legibility especially in low-light or vibration-prone environments in which characters need to be larger.

Because of the limitations on the form size, more forms are needed to provide complete functionality [2] (which leads to web site-style navigation issues). For the picking list a minimum of three forms are needed to cover the following actions:

- 1) Display picking lists and allow selection of one of them.
- 2) Display stock items on the picking list and allow selection of an item.
- 3) Display warehouse location of the item and allow a quantity to be entered.

Fig. 2. Comparison of PDA to desktop form size

C. Colour and sound

The iPaq has a 256 colour capacity which is sufficient to provide a very clear and sharp interface. Good definition of controls is possible through the use of colours, but the screen can be overwhelmed by strong sunlight, and so testing should be carried out in different lighting conditions. Colour can be used to indicate state, but this can have serious drawbacks for users who are colour blind². In order to explore the effects of colour a version of the prototype that used colour to indicate the state of the items was produced and a colour-blind user was asked for his reactions. As anticipated, greatest difficulty was had in discriminating between small red and green objects.

Many of the newer PDA devices have audio capability, the utility of which is dependent on the environment's ambient noise. Reminders and warnings with audio attached to them work well when in a low noise situation, but are more difficult to hear in the open or in noisy places [1]. There are serious ergonomic and safety issues concerning the use of headphones in industrial environments.

D. Messages and status controls

There is no physical space for a status bar on the PDA's forms, and so program status will need to be obvious from the current form. Messages can be provided either in a designated field on the form, or via popup message boxes.

Another feature of a touch screen interface is that there are no mouse-related focus events, such as mouseover or mouseleave. So, the method of providing visual feedback to the user about the status of, for example, a clickable picture such as can be found on a web page cannot be used. This limits the feedback a designer can expect to provide a user on the status of any given control.

E. Scrolling, lists and tree views

It is difficult to use dragging to control a scroll bar on a touch screen. This is especially difficult when the iPaq's Transcriber tool is turned on, as it sometimes interprets the drag as a gesture or line. It is easier to use the clickable components that move the bar, such as the arrow buttons at either end. An associated problem is handling lists of information. Tree views are difficult to manage on small screens, especially if they expand to many nesting levels. It is easier to view a list of information, and navigate via the click scrolling method described above.

²Around 8% of the Caucasian male population and 0.5% of the female population are colour blind; see http://www.internetg.org/newsletter/mar99/accessibility_color_challenged.html

One option for managing large lists is to use the metaphor of a drum or cone of data. The idea is to have the middle of the drum (i.e. the part closest to you) enlarged with the edges of the drum compressed, and therefore providing more lines overall (see Fig. 3).

Stock Code	Description	Quantity
A134	Lamp	10
B654	Wire	1
L635	Extension	98
N762	Blue Lamp	165
F762	Switch	19
H352	Connector	100
I817	Junction box	64

Fig. 3. Drum list view

V. IMPLEMENTATION ISSUES

A. Range

The 802.11b range of 30 metres, although longer than Bluetooth's 10 metres still seems too short given that warehouses can be enormous; indeed, it is in very large warehouses that mobile computing could have the greatest benefit. In practice, the system worked up to 100 metres when the server was in a building and the PDA in a car park. However, the reception disappeared at a range of 30 metres when houses were in the signal path.

B. Disconnections

Careful consideration needs to be made for wireless based applications where disconnections could be frequent. Managed disconnections and automated re-connection without data loss would certainly be a requirement for a real picking list application. There would be no reason to stop picking an order just because the connection with the main network had been lost. In fact, there may be areas of the warehouse where dead spots occur, or the range is too great for the devices.

One model of client server design that would provide distinct advantages is the 'fat client' approach. This places the business logic and presentation components on the client device, ensuring the application can still be used for data capture while a connection to the server is unavailable. Data can be stored locally on the client device, and updates carried out to the server database when connection is re-established. A client server application design based on a distributed presentation layer, where the business logic and data reside on the server and only the presentation components are available on the client, would not suit this application due to the high risk of constant disconnections.

VI. EVALUATION OF THE INTERFACE

A. Visibility of system status

It is important to inform the user what the system is doing at any time. This was difficult given the iPaq's small screen area. In some respects, the system status is not always obvious from the screens in the prototype. For example, the first version of a form showing a list of all stock items on the accounting system caused the system to appear to hang. In fact there was a delay between the extraction of the data from the server and its transmission and subsequent display making it appear to the user that nothing was happening. This was worked around by breaking the transmission into chunks and displaying the chunks as they were received.

This had two effects: users could see the interface being populated and so know the software was still working despite the

screen not responding to input. Secondly, it managed expectations of response time. Many users will prod and poke at the screen if the system appears to be taking too long to respond. Incrementally populating the interface can alleviate frustration.

However, this method increased the overall time for the data transfer considerably. Users rated the faster option as more satisfactory, but this was only after a few uses. On first use of the form they were unsure what was happening when the refresh was based on the method that shows no status. Only after becoming comfortable with the interface did they then prefer the faster method. This is something to consider when designing an interface for use by novice and expert users.

One challenge of interface for the iPaq is its 'always-on' design. Forms that are loaded in an application stay loaded until the application is terminated. Thus, because the main database could have been updated by the accounting product a user could be presented with a form that is out of date. For this reason, the load event of the form is not the best place to put code to initialise the form or synchronise it with the server application.

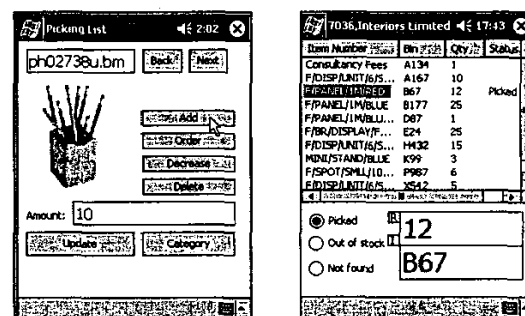


Fig. 4. Alternative picking list representations

B. Picking list interaction

The order of the items on a typical printed picking list would normally be in the most efficient order for the store person to pick the stock. So the prototype allows the user to walk through the warehouse looking for items to pick, presenting each item in the same order as a paper picking list.

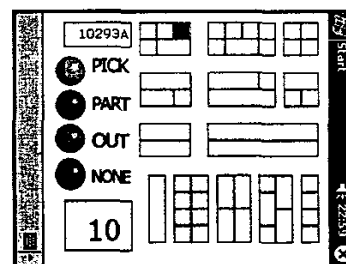


Fig. 5. Warehouse location clues

There were several ways of assisting a store person in this task, either by showing a picture of each item as it was selected or by showing the bin location of the item (see Fig. 4). This was further enhanced by also locating the item on a representation of the

warehouse (Fig. 5). If a context sensitive network were in place, the store person could even be directed to the correct location. Users who had been instructed in the picking list process reported that the warehouse location form was logical and easy to use despite the lack of obvious buttons.

C. Handedness

An area of design for a PDA that does not affect desktop applications so much is the question of user handedness. On a PDA the side of the screen on which controls are positioned can be very important. When a user touches the screen with a finger or stylus part of the display is obscured. If the user is left-handed and the controls are placed on the right, the bulk of the screen will be hidden by the user's hand and this can impact upon the task. On a touch sensitive screen the user's hand may action an obscured control without the user realising. Scrolling becomes slower as the user must move their hand out of the way to see where the current status of the list. Handedness must be considered as a design standard when using touch screens. The prototype was built with the option to switch the usable controls to either side of the screen. Left-handed users reported a preference for this facility.

D. Data entry

One problem everyone had with the interface was with the iPaq's Transcriber input software. For data entry the user must activate the stylus's pen mode before attempting to write in a text box. For form manipulation the pen mode must be disabled. Users often forgot to do this and delays and confusion resulted when going from selecting a stock item (pen off) to changing a quantity or location (pen on). Some users commented that it would be better if the pen mode automatically turned itself on and off as the form context changed. This would have prevented the user from making mistakes on the form, but it is not possible to control the pen software from a VB application.

Some users attempted to write in the spaces provided for the data display. For example, the quantity field is very small (see Fig. 4), but each person tried to write in this box to change the quantity. When questioned they said they thought the text box was where the information was needed, so they presumed they had to write the information in that box. They did not expect the system to read the information from anywhere on the screen and put it into the correct place.³ For people used to filling in paper (and HTML) forms, the data boxes appear to have writing affordances; conversely, a PDA screen superficially resembles a desktop display which does not normally afford writing (keyboards provide the input), yet an iPaq's entire screen surface possesses an actual input affordance.

A possible solution to this affordance/perception imbalance is to make sure fields are large enough and in a convenient screen location to allow writing in. Some users tended always to write in white space, so a reasonably large area of blank space could be left for this purpose.

E. Recognition or recall

Although designed to be easy to use, some forms were so far removed from the norm of desktop interfaces that some users had difficulty in using the application. This mainly occurred when forms were rotated 90° to a landscape orientation (see Fig. 5). The lack of

any of the usual interface components (buttons, sliders, scroll bars) added to initial confusion. However, once the purpose and interaction was explained, which took a few seconds, users then found the form simple and straightforward to use. Even after using the form for just a few minutes, they always remembered the form on subsequent trials (up to a month apart).

F. Minimalist aesthetic

A lot of information can be difficult to read on a small screen, making the interface hard to use. Lists had this effect in the prototype, providing a very compact display area with high information density. The small fonts required to display the volume of data made these forms less popular, until users realised that only these forms gave feedback on the picking lists status. The tasks of reading and updating slowed down with these forms, whilst the large format forms were faster and reportedly easier to use (see Fig. 4).

The larger format forms allowed information not on the list-based form that is helpful to the employee to be displayed. For example, the left-hand form in Fig. 4 includes a picture of the stock item which could be useful to a new employee. Of course, pictures take up a large amount of space compared to text, so the use of pictures should be limited to those areas of the application where they could add value to the process. However, this will not be suitable in all situations. For example, a picture of a 5mm coarse thread bolt may well be indistinguishable from a 6mm coarse thread bolt.

G. Hardware issues

The physical size of the display did not appear to pose a problem, and provided enough screen area for the production of a number of quite different types of form. The screen resolution was sufficient to provide for quite small images and text for those areas of the interface where background controls and information could be provided.

The iPaq's virtual keyboard caused some problems as it takes up a large portion of the screen when activated. It is relatively easy to use, but can cause problems if the field being entered into appears near the lower half of the form. The virtual keyboard can then cover up the field so the user does not know what has been entered.

The reliability and robustness of the device has been acceptable, although no-one dropped it on the floor. There may be a long term problem with heavy use of the touch screen as small scratches can be seen on the surface if the device is used at an angle instead of flat to the user.

With a battery pack and wireless network card the device was quite heavy at 410gms compared to 175gms for the PDA alone. However, users reported it felt more stable with the extra weight of the expansion pack and network card.

H. Limitations of the software

The development environment was a great disappointment. Differences between the desktop version of Visual Basic and the embedded version (e.g. no with...end block support, no enumerations or constants in forms, and reduced functionality in the controls) made the programming task more difficult.

It is intended to develop future versions of the software in C#, which has direct support for the entire range of API interfaces provided in the Pocket PC 2002 operating system.

³ Unlike the Palm OS devices which have a dedicated writing area, the iPaq's Transcriber software allows writing to be done anywhere on the display.

One issue that is not obvious is the device's reliance on third party add-ons, such as the Transcriber and virtual keyboard software. Although this software is supplied with the PDA, it is not an integral part of the operating system. This could have implications for developments based on this functionality being available on the PDA in the future.

1. Other Issues

Users may experience fatigue after extended use. The PDA, with expansion pack and wireless network card weighed 410gms. This could feel very heavy after a few hours use, and might start to cause physiological problems. One possible solution to this is to provide a holder of some sort on the vehicle or trolley being used to hold the picked stock.

Although the iPaq had a clear and bright screen in the office, in certain situations the screen was almost unreadable, such as in very strong sunlight. The larger the interface components, the easier it was to make them out, but some uses may need to consider a filter device to eliminate screen glare and reflection. Even bright fluorescent lighting had a detrimental effect on the visibility of the screen. It is possible to use bright background colours to hide fingerprints and reduce the glare, but not black as this emphasised fingerprint marks. Patterned backgrounds may also help the eye to focus on the screen image instead of reflections.

VII. CONCLUSIONS & FURTHER WORK

In general users preferred to use the picture-based picking list forms and disliked the very dense list-based forms. However, one user preferred the list view as he liked the amount of readily available information. All users were very positive about the landscape forms, and particularly liked the layout, feedback, and use of colour. They found them easy to learn and to use.

The next stage in this work is to build a more comprehensive version of the system that makes good and consistent use of the features that received the most positive feedback. To evaluate these features, alternative versions will be produced that make sole use of the list view, large format, and landscape format forms, and a formal experiment conducted to allow quantitative analysis of the interfaces.

Without exception, users attempted to write data into either the forms' data fields or into areas of white space. Therefore, the next version of the system will incorporate forms that include areas of white space set aside for data entry or larger text boxes and this aspect will be formally compared for performance and user preference against forms without these features.

User feedback on the prototype system supports the idea that mobile devices can be used to carry out business functions that have previously been restricted to desktop workstations. Forms containing large amounts of data that work well in desktop environments were not well received by users of the prototype. This supports the idea that simply converting a desktop application to run on a mobile device will not lead to maximum benefit. To take full advantage of mobile devices with touch screen interfaces business processes should be thoroughly analysed and re-designed for the mobile environment. It is unlikely that entire desktop applications can be successfully migrated to today's handheld mobile devices. For example, due to the restrictions of large volume data entry, it

would not seem appropriate to transfer invoice entry to a PDA. Of course, keyboard attachments can be connected to handheld devices to allow volume input, but doing this begins to remove the advantages of a handheld device: if a mobile platform with a keyboard is needed then a laptop/notebook computer would be a better choice. These ideas can be explored in further formal studies that will compare performance of common tasks on both mobile and desktop systems.

The PDA with its touch screen interface presents the opportunity to provide functionality at the point best suited to its use. Applications such as picking lists or the restaurant table waiting systems would seem to be ideal for this type of mobile device, potentially saving hours of unnecessary staff movements. Further, the interface can be designed in such a way as to expand the functionality of relatively simple tasks to provide real benefits to new employees or the disabled, supplementing their knowledge or physiology to allow them to complete a larger number of tasks with minimal training.

The range of wireless network technologies remains problematic and it may be worth exploring the potential of the GPRS and 3G mobile telephony standards for this type of application.

Using the lessons learnt from this heuristic pilot study new versions of the picking list (and other accounting functions) will be built and full user studies conducted.

REFERENCES

- [1] Brewster, S.A., *Overcoming the Lack of Screen Space on Mobile Computers*. Department of Computing Science Technical Report TR-2001-87, April. Glasgow University.
- [2] Chittaro, L. and Dal Cin, P., Evaluating Interface Design Choices on WAP Phones: Single-choice List Selection and Navigation Among Cards. in *Proc Mobile HCI 2001: Third International Workshop on Human Computer Interaction with Mobile Devices*, M.D. Dunlop and S.A. Brewster, eds. 2001. Lille.
- [3] Cooper, A., *The Inmates Are Running The Asylum*. 1999, Indianapolis: Macmillan Computer Publishing.
- [4] Iacucci, G., Bridging Observation and Design in Concept Development for Mobile Devices. in *Proc HFWeb 2001: Seventh Human Factors and Web Conference 2001*. Madison, Wisconsin.
- [5] Norman, D.A. and Draper, S.W., *User-Centred System Design: New Perspectives on Human-Computer Interaction*. 1986, Hillsdale: Lawrence Erlbaum Associates.
- [6] Rist, T., Using Mobile Communication Devices to Access Virtual Meeting Places. in *Proc. Mobile HCI 1999: Second Workshop on Human Computer Interaction with Mobile Devices*, M.D. Dunlop and S.A. Brewster, eds. 1999. Edinburgh.
- [7] Vanderdonckt, J., Florins, M., and Oger, F., Model-Based Design of Mobile User Interface. in *Proc. Mobile HCI 2001: Third International Workshop on Human Computer Interaction with Mobile Devices*, M.D. Dunlop and S.A. Brewster, eds. 2001. Lille.